

MR DBA

Oracle, Mysql e Sqlserver

Configurando HUGEPAGE em Banco de Dados Oracle 11g

JUN 17

Publicado por [mrochadba](#)

Hugepage em Oracle 11g

Muitos DBAs tem o conceito errado de que HugePages é um método usado exclusivamente para possibilitar o aumento da SGA em servidores Linux 32 bits, e a idéia do documento é apresentar os benefícios em se implementar o método em máquinas x86-64 bem como demonstrar o procedimento necessário para ativar a funcionalidade em bancos de dados Oracle 11g.

Introdução

HugePages é uma funcionalidade presente no kernel 2.6 que permite que o sistema operacional trabalhe com páginas de até 2MB (x86-64) ao invés do tamanho default de 4KB. Esta funcionalidade é bastante útil quando trabalhamos com servidores com grande quantidade de memória.

Os Sistemas Operacionais Linux possuem uma característica muito interessante chamada de HugePages, que quando usada por um banco Oracle oferece grandes ganhos de performance.

HugePages consiste em tornar os blocos de alocação de memória muito maiores do que os 4KB padrão. No Linux os HugePages possuem 2MB. Outra vantagem é que os HugePages são sempre alocados na memória física e nunca vão para a área de swap. Na verdade esses blocos de memória sequer são considerados como candidatos a ir para swap.

Benefícios

1. Não utiliza swap, portanto evita-se o overhead existente no mecanismo de page-in/page-out
2. Elimina a busca por páginas em swap.
3. O processo kswapd será muito requisitado caso tenha que gerenciar uma grande quantidade

de páginas de memória e certamente utilizará uma grande quantidade de processamento da máquina. **Quando HugePages é implementado, este processo não é envolvido no gerenciamento.**

Roteiro básico

Para implementar HugePages em Oracle 11g é necessário desabilitar a funcionalidade de gerenciamento automático de memória, que torna a alocação de memória SGA dinâmica.

Além disso, os seguintes passos, já conhecidos de versões anteriores, são necessários na implementação:

1. Verificar a compatibilidade:

```
root$ grep Huge /proc/meminfo
```

```
HugePages_Total: 0 — — — zerado (hugepage não funcionando ainda)
```

```
HugePages_Free: 0
```

```
HugePages_Rsvd: 0
```

```
Hugepagesize: 2048 kB
```

2. Iniciar a instância Oracle

```
oracle$ sqlplus / as sysdba
```

```
SQL*Plus: Release 11.2.0.1.0 Production on Thu May 6 12:05:25 2010
```

```
Copyright (c) 1982, 2009, Oracle. All rights reserved.
```

```
Connected to:
```

```
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 – Production
```

```
With the Partitioning, OLAP, Data Mining and Real Application Testing options
```

```
SQL> select open_mode from v$database;
```

```
OPEN_MODE
```

```
-----
```

```
READ WRITE
```

3. Verificar se o gerenciamento automático de memória do Oracle está desabilitado:

```
SQL> show parameter memory_target
```

```
NAME TYPE VALUE
```

```
-----
```

```
memory_target big integer 0
```

4. Definir memlock

Para implementar HugePages no Linux x86-64, a Oracle recomenda que o valor de memlock seja maior que o tamanho da SGA. Para uma SGA de 5 GB, coloque sempre superior o valor mais nada impede de configurar com o mesmo valor, mais é sempre bom configurar com uma folga.

```
root$ grep memlock /etc/security/limits.conf
```

```
# – memlock – max locked-in-memory address space (KB)
```

```
oracle hard memlock 5243000
```

```
oracle soft memlock 5243000
```

5. Calcular nr_hugepages utilizando o script hugepages_settings.sh oferecido pela Oracle no seguinte note:

***Como criar o script hugepages_settings.sh no final da documentação.**

Esse script lhe permite achar o valor exato para o parâmetro – nr_hugepages

Shell Script to Calculate Values Recommended Linux HugePages / HugeTLB Configuration [ID 401749.1]

```
oracle$ ./hugepages_settings.sh
```

This script is provided by Doc ID 401749.1 from My Oracle Support (<http://support.oracle.com> (<http://support.oracle.com>)) where it is intended to compute values for

the recommended HugePages/HugeTLB configuration for the current shared memory segments. Before proceeding with the execution please make sure that:

- * Oracle Database instance(s) are up and running

- * Oracle Database 11g Automatic Memory Management (AMM) is not setup (See Doc ID 749851.1)

- * The shared memory segments can be listed by command:

```
# ipcs -m
```

Press Enter to proceed...

Recommended setting: vm.nr_hugepages = 105

6. Alterar parametro de kernel conforme valor recomendado no procedimento anterior:

```
root$ grep hugepages /etc/sysctl.conf
```

```
root$ echo "vm.nr_hugepages = 105">> /etc/sysctl.conf
```

```
root$ sysctl -p
```

7. Reinicie a instância Oracle

```
SQL> shutdown immediate
```

Database closed.

Database dismounted.

ORACLE instance shut down.

```
SQL> startup
```

ORACLE instance started.

Total System Global Area 431058944 bytes

Fixed Size 1336960 bytes

Variable Size 213911936 bytes

Database Buffers 209715200 bytes

Redo Buffers 6094848 bytes

Database mounted.

Database opened.

8. Verifique o funcionamento

```
root$ grep Huge /proc/meminfo
```

HugePages_Total: 1

HugePages_Free: 1

HugePages_Rsvd: 0

Hugepagesize: 4096 kB

Caso a configuração não se efetive, pode ser necessário o reboot do servidor.

Roteiro a ser seguido

Configuração

A primeira coisa a ser feita para a configuração é identificar quanta quantidade de memória você necessita para ser usada como HugePages. No caso de uma instância Oracle você deve alocar uma quantidade de memória suficiente para acomodar todo o seu SGA. Para isso use a fórmula abaixo.

$$Z = Y / X$$

Onde:

- X: Tamanho do HugePage em KB (no caso do Linux será sempre 2048);
- Y: Tamanho do seu SGA também em KB (se utilizar o valor do PFILE, divida por 1024);
- Z: Quantidade de HugePages necessárias.

Obs.: Uma dica é sempre configurar um valor um pouco maior do informado na fórmula. Eu normalmente somo 8 no resultado encontrado.

Agora que você já sabe quanto de HugePages configurar, vamos colocar a mão na massa!

Servidor – ha1

Achando o Hugepage

Para achar o valor de Hugepage pode ser utilizado 2 metodos, o metodo acima $Z=Y/X$ acima setado ou metodo utilizado pelo script da Oracle, o qual vamos utilizar.

./hugepages_settings.sh o qual eu recomendo.

Para executar este script é necessário que a instancia esteja no ar, sem gerenciamento automático de memória.

Para executar o script e necessário que já tenha em mente o quanto de memória deseja usar: No meu caso eu tenho um servidor com 94G de memória mais vou configurar 35G.

35G de 94G.

35G -- ok

A instância oracle já deve ser configurada para 35G de SGA.

Configurando o kernel linux para 35G.

Configurando para 35G — vi /etc/sysctl.conf

kernel.shmmax = 37580963840

Controls the maximum number of shared memory segments, in pages – **pagesize = 4096**

kernel.shmall = **shmmax** / pagesize => 37580963840 / 4096 .:

kernel.shmall = 9175040

oracle\$ /sbin/sysctl -p # carregar os valores alterados nas variáveis

Configurando o Banco de Dados:

No Oracle o que deve ser feito é desabilitar o gerenciamento automático de memória (AMM).

Essa configuração não é requerida na versão 10g, apenas na 11g. Mas é recomendado que seja aplicada em ambos pois o ganho de performance pode ser significativo. Isso ocorre pois o gerenciamento automático de memória pode, em algumas situações, consumir muitos recursos e causar problemas de performance, dessa forma a recomendação geral é que utilize tamanhos pre-definidos para os pools no SGA.

Para desabilitar o AMM basta configurar o parâmetro abaixo para “0” com o seguinte comando:

```
SQL> alter system set sga_target=0 scope=spfile;  
SQL> alter system set memory_max_target=0 scope=spfile;  
SQL> alter system set memory_target=0 scope=spfile;  
SQL> alter system set lock_sga=FALSE scope=spfile;
```

Configuração de Memória Manual.

De 35G separado para o Banco de Dados Oracle, os quais serão configurados para HUGE PAGES no sistema operacional, o Banco de Dados Oracle terá seu startup com 25G máximo, mais internamente seus outros parâmetros o deixará com 20G aproximadamente, podendo depois aumentar até os 25G

```
SQL> alter system set sga_max_size =25G scope=spfile;
```

```
sga_max_size big integer 25G  
memory_max_target big integer 0  
memory_target big integer 0  
sga_target big integer 0  
lock_sga boolean FALSE
```

Quando se está configurado o Banco de Dados para gerenciamento de memória manual, é necessário setar os parâmetros abaixo:

```
SQL> alter system set pga_aggregate_target =3G scope=spfile;
SQL> alter system set db_cache_size =7424M scope=spfile;
SQL> alter system set shared_pool_size =7G scope=spfile;
SQL> alter system set java_pool_size =832M scope=spfile;
SQL> alter system set streams_pool_size = 768M scope=spfile;
SQL> alter system set large_pool_size =1G scope=spfile;
```

```
pga_aggregate_target big integer 3G
db_cache_size big integer 7424M
shared_pool_size big integer 7G
java_pool_size big integer 832M
log_buffer integer 69206016 -- não alterar (default)
streams_pool_size big integer 768M
large_pool_size big integer 1G
```

log_buffer integer 69206016 — valor default não alterar

Após isso será necessário reiniciar a instância para aplicar a configuração, pois trata-se de um parâmetro estático.

Os parâmetros foram setados para comportar dentro de um SGA de 25G, deixando uma folga para aumentos futuros, caso tenha que aumentar novamente, abaixo de 35G, basta setar um valor superior para `sga_max_size` que seja abaixo de 35G e setar os outros parâmetros tal que não seja superior a SGA.

```
SQL> shutdown immediate
...
SQL> startup
```

Após colocar o banco em gerenciamento de memória manual, já podemos começar a configuração do HugePage na máquina.

Configurando o Linux

O primeiro passo é editar o arquivo `/etc/security/limits.conf` e definir a quantidade máxima de memória que o usuário oracle pode alocar:

```
oracle soft memlock xxxxxxxx
oracle hard memlock xxxxxxxx
```

Servidor – ha1

```
vi /etc/security/limits.conf
```

```
# Setando o memlock para 35G para um sga de 35G
```

```
oracle hard memlock 36700160
oracle soft memlock 36700160
```

```
[root@hall:~]# grep memlock /etc/security/limits.conf
```

```
# – memlock – max locked-in-memory address space (KB)
```

```
oracle hard memlock 36700160
oracle soft memlock 36700160
[root@hall:~]#
```

O passo seguinte é editar o arquivo `/etc/sysctl.conf` e adicionar os seguintes parâmetros:

- `vm.nr_hugepages = <Número de HugePages>`
- `vm.hugetlb_shm_group =`

Para verificar o ID do grupo principal do usuário oracle utilize o comando `id`:

```
# id oracle
uid=500(oracle) gid=501(oinstall) groups=501(oinstall),500(dba)
```

No exemplo acima o ID é **501**.

Servidor – ha1

Com o banco de Dados no Ar e a base de dados estando configurado para gerenciamento de memória manual, podemos executar o script abaixo para achar o valor de HugePage configurados para 35G.

Logado como usuário oracle: execute o script – **hugepages_settings.sh**

```
oracle$ ./hugepages_settings.sh
This script is provided by Doc ID 401749.1 from My Oracle Support
(http://support.oracle.com (http://support.oracle.com)) where it is intended to compute values
for
the recommended HugePages/HugeTLB configuration for the current shared
memory segments. Before proceeding with the execution please make sure
that:
* Oracle Database instance(s) are up and running
* Oracle Database 11g Automatic Memory Management (AMM) is not setup
(See Doc ID 749851.1)
* The shared memory segments can be listed by command:
# ipcs -m
Press Enter to proceed...
Recommended setting: vm.nr_hugepages = 12808
[root@ha1:~]# id oracle
```

```
uid=501(oracle) gid=501(oinstall) groups=501(oinstall),502(dba),503(oper)
[root@ha1:~]#
```

Grupo oracle – 501

Abra o arquivo `/etc/sysctl.conf` e adicione as linhas abaixo:

```
vi /etc/sysctl.conf

vm.nr_hugepages = 12808
vm.hugetlb_shm_group=501
```

Depois de configurar esses parâmetros reinicie o S.O. para que as configurações entrem em vigor. Para validar use os seguintes comandos:

```
[root@ha1:~]# cat /proc/meminfo
MemTotal: 99052404 kB
MemFree: 7265892 kB
Buffers: 730096 kB
Cached: 61531308 kB
SwapCached: 0 kB
Active: 27687168 kB
Inactive: 34843268 kB
VmallocTotal: 34359738367 kB
VmallocUsed: 479612 kB
VmallocChunk: 34307793532 kB
HardwareCorrupted: 0 kB
AnonHugePages: 69632 kB
HugePages_Total: 12808 KB -- já não está mais zero
HugePages_Free: 6783 KB
HugePages_Rsvd: 6776 KB
HugePages_Surp: 0 KB
Hugepagesize: 2048 kB
DirectMap4k: 5604 kB
DirectMap2M: 2078720 kB
DirectMap1G: 98566144 kB
[root@ha1:~]#
```

Verifique se o item de *HugePages_Total* está com o valor configurado.

Para validar o limite de alocação de memória do usuário oracle, logue-se com ele e use o comando abaixo:

```
# ulimit -l
```

Verifique se o valor retornado é igual ao configurado.

Para Finalizar a configuração é necessário alterar o parâmetro `kernel.shmall` para o novo parâmetro onde o `pagesize` agora é 2M

```
shmax / pagesize = shmall          shmax => ((35*1024)*1024)
```

```
pagesize => 2M
```

```
# Como o tamanho da pagina agora e 2M pela configuracao do hugepage
# entao shmax / pagesize = shmall => ((35*1024)*1024)/2
```

`kernel.shmall = 18350080` -- valor que deve ser setado para `shmall`

Depois disso, baixe o banco de dados, e execute o comando abaixo para o novo parâmetro entrar em vigor, depois start a base de dados novamente.

* Shutdown banco de dados

execute o comando como root.

```
/sbin/sysctl -p
```

* Startup no banco de dados.

Para saber se o Oracle iniciou utilizando as páginas em HugePages utilize novamente o comando “cat /proc/meminfo” e verifique a diferença entre os valores dos itens *HugePages_Total* e *HugePages_Free*. Se o valor de free for menor que o de total a sua instância está utilizando os HugePages, caso contrário, valide novamente as configurações e reinicie novamente a instância Oracle.

Fazendo o arquivo **hugepages_settings.sh**

—

No sistema operacional crie um arquivo chamado hugepages_settings.sh e cole o seguinte conteúdo dentro dele.

```
#!/bin/bash
#
# hugepages_settings.sh
#
# Linux bash script to compute values for the
# recommended HugePages/HugeTLB configuration
#
# Note: This script does calculation for all shared memory
# segments available when the script is run, no matter it
# is an Oracle RDBMS shared memory segment or not.
#
# This script is provided by Doc ID 401749.1 from My Oracle Support
# http://support.oracle.com (http://support.oracle.com)
#
# Welcome text
echo "
This script is provided by Doc ID 401749.1 from My Oracle Support
(http://support.oracle.com (http://support.oracle.com)) where it is intended to compute values
for
the recommended HugePages/HugeTLB configuration for the current shared
memory segments. Before proceeding with the execution please make sure
that:
* Oracle Database instance(s) are up and running
* Oracle Database 11g Automatic Memory Management (AMM) is not setup
(See Doc ID 749851.1)
* The shared memory segments can be listed by command:
# ipcs -m

Press Enter to proceed..."

read
```

```
# Check for the kernel version
```

```
KERN=`uname -r | awk -F. '{ printf("%d.%d\n",$1,$2); }`
```

```
# Find out the HugePage size
```

```
HPG_SZ=`grep Hugepagesize /proc/meminfo | awk '{print $2}`
```

```
# Initialize the counter
```

```
NUM_PG=0
```

```
# Cumulative number of pages required to handle the running shared memory segments  
for SEG_BYTES in `ipcs -m | awk '{print $5}' | grep "[0-9][0-9]*`
```

```
do
```

```
MIN_PG=`echo "$SEG_BYTES/($HPG_SZ*1024)" | bc -q`
```

```
if [ $MIN_PG -gt 0 ]; then
```

```
NUM_PG=`echo "$NUM_PG+$MIN_PG+1" | bc -q`
```

```
fi
```

```
done
```

```
RES_BYTES=`echo "$NUM_PG * $HPG_SZ * 1024" | bc -q`
```

```
# An SGA less than 100MB does not make sense
```

```
# Bail out if that is the case
```

```
if [ $RES_BYTES -lt 100000000 ]; then
```

```
echo "*****"
```

```
echo "*** ERROR ***"
```

```
echo "*****"
```

```
echo "Sorry! There are not enough total of shared memory segments allocated for  
HugePages configuration. HugePages can only be used for shared memory segments  
that you can list by command:
```

```
# ipcs -m
```

```
of a size that can match an Oracle Database SGA. Please make sure that:
```

```
* Oracle Database instance is up and running
```

```
* Oracle Database 11g Automatic Memory Management (AMM) is not configured"
```

```
exit 1
```

```
fi
```

```
# Finish with results
```

```
case $KERN in
```

```
'2.4') HUGETLB_POOL=`echo "$NUM_PG*$HPG_SZ/1024" | bc -q`;
```

```
echo "Recommended setting: vm.hugetlb_pool = $HUGETLB_POOL" ;;
```

```
'2.6') echo "Recommended setting: vm.nr_hugepages = $NUM_PG" ;;
```

```
*) echo "Unrecognized kernel version $KERN. Exiting." ;;
```

```
esac
```

```
# End
```

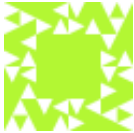
O arquivo deve possuir permissão para execução;

```
rw-r-xr-x 1 root root 2428 Apr 20 11:08 hugepages_settings.sh
```

You May Like



- 1. [8](#)
[Mysterious Photos That Have Yet to be Explained](#) 3 weeks ago
[answers.com](#) [Answers](#) [Answers.com](#)
(sponsored)



Sobre mrochadba

Database Administrator

[Visualizar todos os posts por mrochadba »](#)

Publicado em 17/06/2015, em [Uncategorized](#). Adicione o [link](#) aos favoritos. [Deixe um comentário](#).

○ **Deixe um comentário**

○ **Comentários 1**

[Crie um website ou blog gratuito no WordPress.com. O tema Mystique.](#)

◎ Seguir

Seguir “MR DBA”

Crie um site com WordPress.com